BAB I
PENDAHULUAN

I.1. Mikrokontroller AT89C51

Sebuah mikrokontroller selain memiliki CPU juga dilengkapi memori dan input output yang merupakan kelengkapan sebagai sistem minimum mikrokomputer sehingga sebuah mikrokotroller dapat dikatakan sebagai mikrokomputer dalam kepingan tunggal (Single Chip Mikrokomputer) yang dapat berdiri sendiri/stand alone.

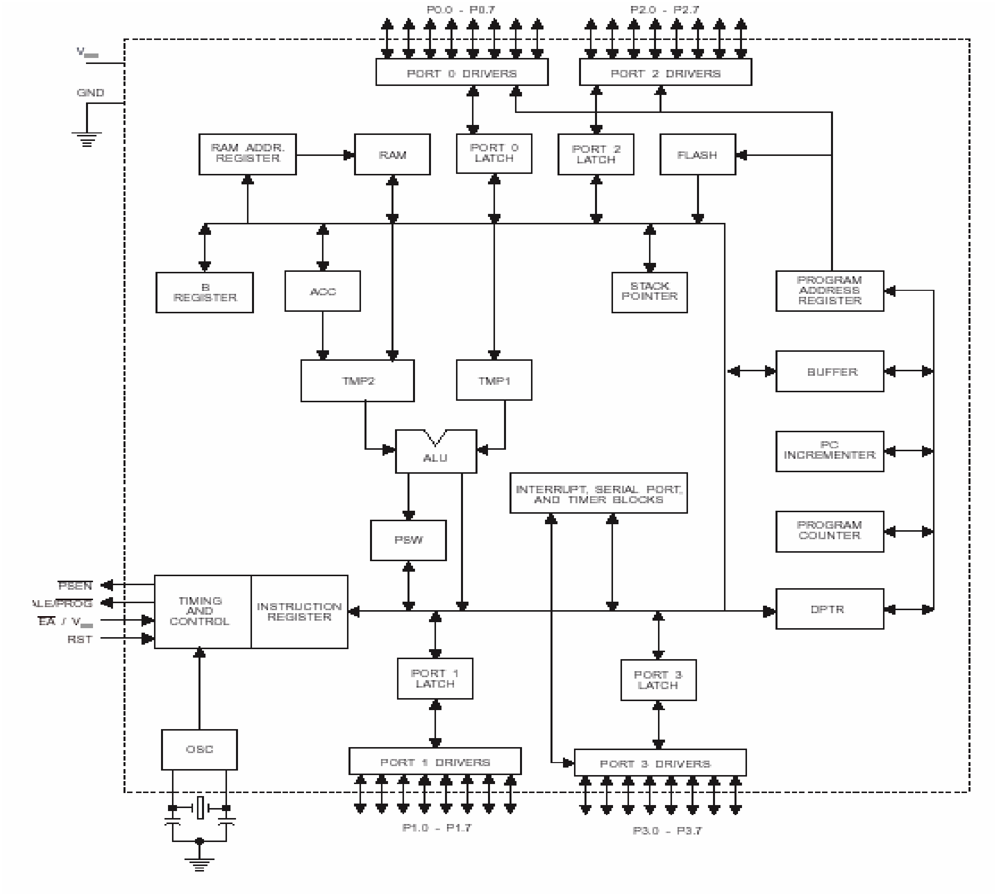
Mikrokontroller AT89C51 adalah mikrokontroller ATMEL yang kompatibel dengan mikrokontroller keluarga MCS-51, yang membutuhkan data rendah dan memiliki performance yang tinggi dan merupakan mikrokontroller 8 bit yang dilengkapi 4 Kbyte EEPROM (Electrical Erasable Programmable Read Only Memory) dan 128 byte RAM internal, program memory dapat diprogram ulang dalam sistem atau menggunakan Programmer Nonvolatile Memory Konvensional.

Dalam sistem mikrokontroller terdapat 2 hal yang mendasar, yaitu perangkay keras (hardware) dan perangkat lunak (software) yang keduanya saling berkaitan dan mendukung satu sama lain.

I.1.1 Perangkat Keras Mikrokontroller AT89C51

Mikrokontroller AT89C51 memiliki :

- Sebuah CPU (Central Processing Unit).
- RAM (Randaom Access Memory).
- EEPROM.
- Empat buah Programmable Port I/O, masing-masing terdiri atas 8 jalur I/O yang dapat diprogram dan dapat dialamati per-bit/bit addressible.
- Dua buah Timer/Counter 16 bit.
- Sumber Interrupt.
- Port Komunikasi Serial.
- Osilator dan Clock.



Gambar 1-1

Blok Diagram Arsitektur Mikrokontroller AT89C51

I.1.2 Arsitektur AT89C51

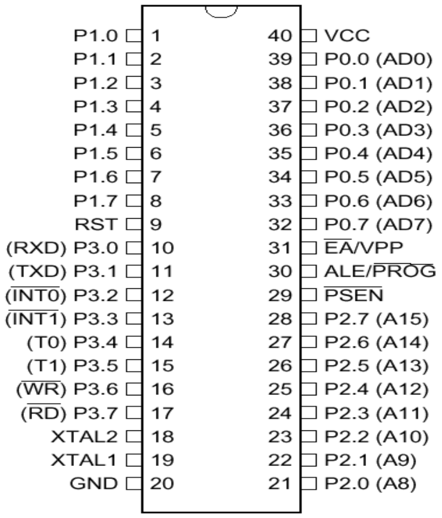
Arsitektur mikrokontroller AT89C51 adalah sebagai berikut :

1. CPU (Central Prosesing Unit) 8 bit dengan register A (Accumulator) dan B.
2. 16 bit program counter (PC) dan data pointer (DPTR).
3. 8 bit program status word (PSW).
4. 8 bit stack pointer (SP).
5. 128 byte internal RAM.
6. 4 Bank Register, masing-masing berisi 4 register :
   a. 16 byte yang dapat dialamati pada bit level.
   b. 80 byte general purpose memori data.
7. 32 pin Input/Output, tersusun atas P0 – P3, masing-masing 8 bit.
8. 2 buah 16 bit Timer Counter.
9. Receiver / Transmitter data serial Full Duplex : SBUF.
10. Control Register, yaitu : TCON, TMOD, CSON, PCON, IP dan IE.

11. 5 buah sumber intrupt yaitu :
    a. 2 buah interupt eksternal.
    b. 2 buah interupt pewaktu.
    c. 1 buah interupsi serial.

I.1.3 Konfigurasi Pena-pena Mikrontroller AT89C51.

Mikrokontroller AT89C51 terdiri dari 40 pin dengan konfigurasi sebagai berikut :



Gambar 1-2

Konfigurasi Pin Mikrokontroller AT89C51

Adapun fungsi dari tiap-tiap pin dari mikrokontroller AT89C51 adalah sebagai berikut :

Nomor Pin	**Nama Pin**	**Fungsi Alternatif**	**Fungsi**
20	GND	-	Ground.
40	Vcc	-	Power Supply.

32-39	P0.7-P0.0	-	Port I/O dua arah(bidirectional), dikonfigurasikan sebagai multipleks bus alamat rendah/LSB (A0-A7) dan data selama pengaksesan program memory dan data memory eksternal.
1-8	P1.0-P1.7	-	Port I/O dua arah (bidirectional) dengan internal pull-up.
21-28	P2.0-P2.7	-	Port I/O dua arah (bidirectional) dengan internal pull-up, mengeluarkan address tinggi (MSB) selama pengambilan (fetch) program memory eksternal dan selama pengaksesan data memory port 2 mengeluarkan P@ SFR (Special Function Register). Menerima address tinggi dan beberapa sinyal kontrol selama pemrograman dan verifikasi.
10-17	P3.0-P3.7	-	Port I/O dua arah (bidirectional) dengan internal pull-up.Sedangkan sebagai fungsi khusus mempunyai keterangan sebagai berikut :
10	P3.0	RXD	Port Serial Input.
11	P3.1	TXD	Port Serial Output.
12	P3.2	INT0	Port Eksternal interupt 0 (masukan interupsi 0).
13	P3.3	INT1	Port Eksternal interupt 1 (masukan interupsi 1).

14	P3.4	T0	Port Eksternal timer 0 input (masukan dari pewaktu/pencacah 0).
15	P3.5	T1	Port Eksternal timer 1 input (masukan dari pewaktu/pencacah 1).
16	P3.6	WR	Sinyal penulisan data luar.
17	P3.7	RD	Sinyal pembacaan memori data luar.
9	RST	-	Pin Input yan aktif tinggi (high), pin ini aktif tinggi selama siklus mesin, ketika osilator bekerja akan mereset peralatan.
30	ALE	PROG	Pulsa input ALE digunakan untuk proses "latching" byte address rendah/LSB (A0-A7) selama pengaksesan ke eksternal memory. Pin ini juga digunakan untuk timing eksternal atau penge-clockan.
29	PSEN	-	Pin ini berfungsi pada saat mengeksekusi program yang terletak pada memori eksternal. PSEN akan aktif dua kali setiap cycle.

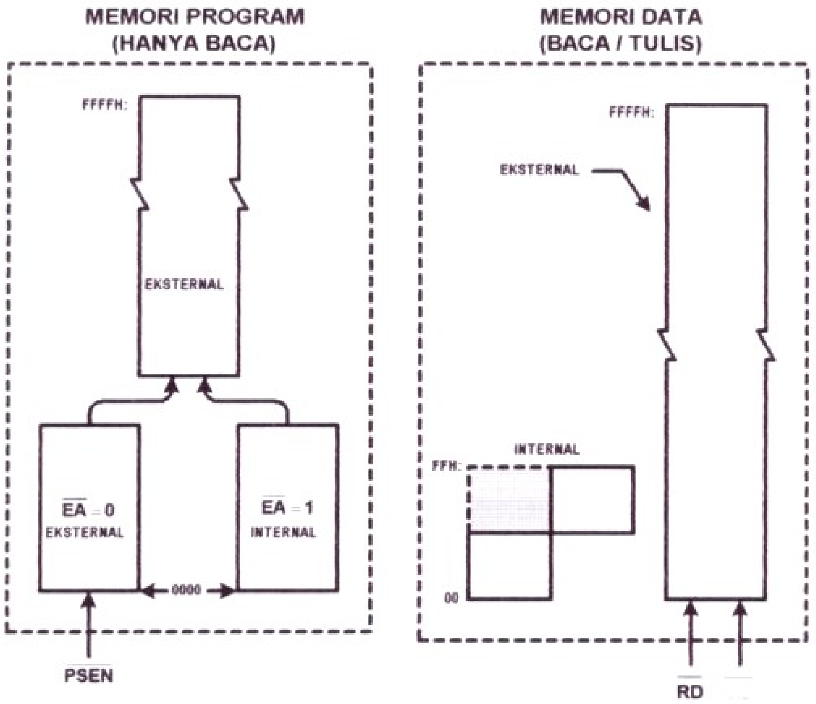
31	EA	VP	Pada kondisi low, pin ini akan berfungsi sebagai EA yaitu mikrokontroler akan menjalankan program yang ada pada memori eksternal setelah sistem di-reset. Jika berkondisi high, pin ini akan berfungsi untuk menjalankan program yang ada pada memori intrnal. Pada saat flash programming pin ini akan mendapat tegangan 12 volt (VP).
19	XTAL1	-	Input Osilator dan dihubungkan, dengan kristal bila menggunakan osilator internal.
18	XTAL2	-	Output osilator dan dihubungkan dengan kristal bila menggunakan osilator internal.

Tabel 1-1

Fungsi Tiap Pin Mikrokontroller AT89C51

I.1.4 Organisasi Memory

Semua produk mikrokontroller Flash AT89C51 Atmel memiliki ruang alamat memori data dan program yang terpisah, sebagaimana yang ditunjukkan pada gambar 2-12. Pemisahan memori program dan data tersebut membolehkan memori data diakses dengan alamat 8 bit, sehingga dapat dengan cepat dan dengan mudah disimpan dan dimanipulasi oleh CPU 8 bit. Namun demikian, alamat memori data 16 bit bisa juga dihasilkan melalui register DPTR (Data Pointer Register).

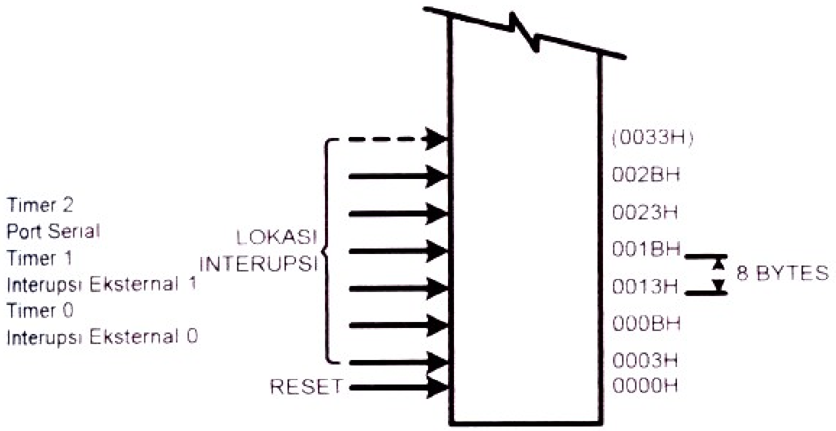


Gambar 1-3

Struktur Memori

I.1.5 Memori Program

Pada gambar 2-17 ditunjukkan pemetaan bagian bawah dari memori. Setelah reset, CPU segera mengerjakan program mulai dari lokasi 0000H.



Gambar 1-4

Memori Program

Sebagaimana ditunjukkan pada gambar 2-75, masing-masing interupsi disimpan pada lokasi yang sudah tetap dalam memori program. Sebuah interupsi menyebabkan CPU melompat ke lokasi interupsi yang

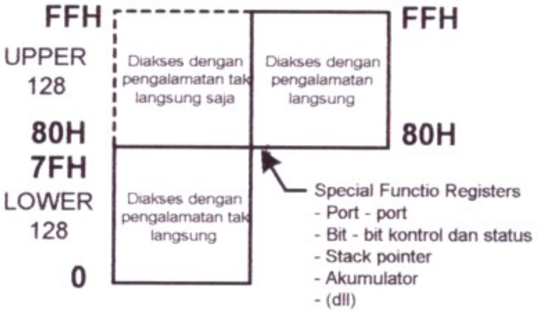
bersangkutan, yaitu letak subrutin layanan interupsi tersebut. Misalnya external interupt 0 disimpan pada lokasi 0003H, jika interupsi ini digunakan maka layanan rutin interupsi ini harus dituliskan pada lokasi ini, jika tidak maka lokasi tersebut dipakai sebagai memori program serbaguna (untuk keperluan lain).

Lokasi-lokasi layanan interupsi tersebut menempati lokasi-lokasi dengan jarak 8 byte. 0003H untuk eksternal interupt 0, 000BH untuk timer 0, 0013H untuk eksternal interupt 1, dan seterusnya. Jika seluruh rutin layanan interupsi sangat pendek (kurang dari 8 byte), maka seluruh rutin bisa disimpan pada lokasi yang bersangkutan (sesuai dengan interupsi yang digunakan). Jika terlalu panjang (lebih dari 8 byte) maka harus digunakan suatu perintah lompat ke lokasi rutin interupsi yang sebenarnya (di lokasi dalam memori program).

Alamat-alamat yang paling bawah dari memori program dapat berada dalam flash on-chip maupun eksternal. Untuk melakukan hal ini dilakukan pengkabelan pada pin EA atau External Access ke Vcc atau ke GND (akses eksternal) sesuai dengan kebutuhan.

I.1.6 Memori Data

Memori data internal dipetakan seperti pada gambar 2-18. Ruang memorinya dibagi menjadi 3 blok, yaitu sebagai lower 128, upper 128, dan ruang SFR (spesial function register).

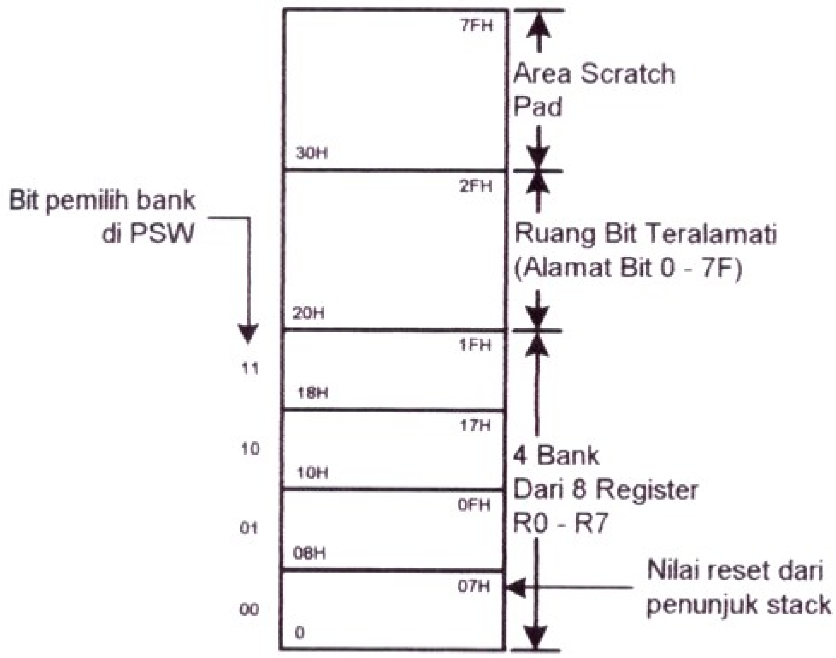


Gambar 1-5

Memori Data Internal

Bagian bawah dari 128 byte RAM dipetakan seperti terlihat pada gambar 2-19. Tiga puluh dua byte paling bawah dikelompokkan dalam 4 bank (8

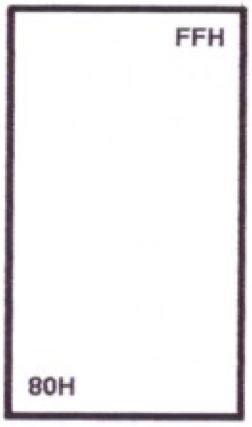
register) yaitu R0 sampai dengan R7. Dua bit PSW (Program Status Word) memilih register bank yang digunakan.



Gambar 1-6

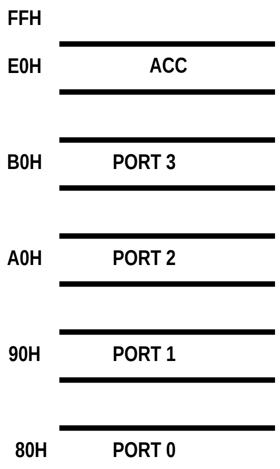
Bagian Bawah 128 byte RAM Internal

Untuk operasi AT89C51 yang tidak menggunakan alamat internal RAM (00H-7FH) dilakukan oleh SFR yang beralamat 80H - FFH seperti yang ditunjukkan pada gambar 2-17, tetapi tidak semua address tersebut digunakan sebagai SFR, hanya address tertentu seperti yang dijelaskan pada tabel 2 – 8.



Gambar 1-7

Bagian Atas 128 byte RAM Internal



Gambar 1-8

Ruang Special Function Register

Simbol	Nama	Alamat
ACC	Akumulator	E0H
B	B Register	F0H
PSW	Program Status Word	D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 16 Bit	
	DPTL Byte rendah	82H
	DPTH Byte tinggi	83H
P0	Port 0	
P1	Port 1	
P2	Port 2	
P3	Port 3	
IP	Interrupt Priority Control	
IE	Interrupt Enable Control	
TMOD	Timer/Counter Mode Control	
TCON	Timer/Counter Control	
TH0	Timer/Counter 0 High byte	
TL0	Timer/Counter 0 Low byte	
TH1	Timer/Counter 1 High byte	

TL1	Timer/Counter 1 Low byte	
SCON	Serial Control	
SBUF	Serial Data Buffer	
PCON	Power Control	

Tabel 1-2

Special Function Register

I.1.7 Metode Pengalamatan

Metode pengalamatan pada mikrokontroller AT89C51 adalah sebagai berikut :

1. Pengalamatan Langsung.

   Pengalamatan langsung dilakukan dengan memberikan nilai ke suatu register secara langsung yaitu dengan menggunakan tanda pagar (#).

2. Pengalamatan Tak Langsung.

   Pada pengalamatan tak langsung, instruksi menunjukkan suatu register yang isinya adalah alamat dari suatu operand, eksternal dan internal RAM dapat dialamati secara tidak langsung. Register alamat untuk data dengan lebar alamat 8 bit dapat berupa R0 dan R1 yang digunakan untuk memilih bank register atau stack pointer. Register alamat untuk data dengan lebar 16 bit digunakan data pointer DPTR. Untuk melaksanakan proses pengalamatan tak langsung digunakan simbol at (@).

3. Pengalamatan Bit (Direct Bit Addressing).

   Pengalamatan bit merupakan penunjukan alamat lokasi bit baik dala RAM internal (byte 32 sampai 47) atau bit perangkat keras. Untuk melakukan pengalamatan bit digunakan simbol titik (.).

4. Pengalamatan Kode

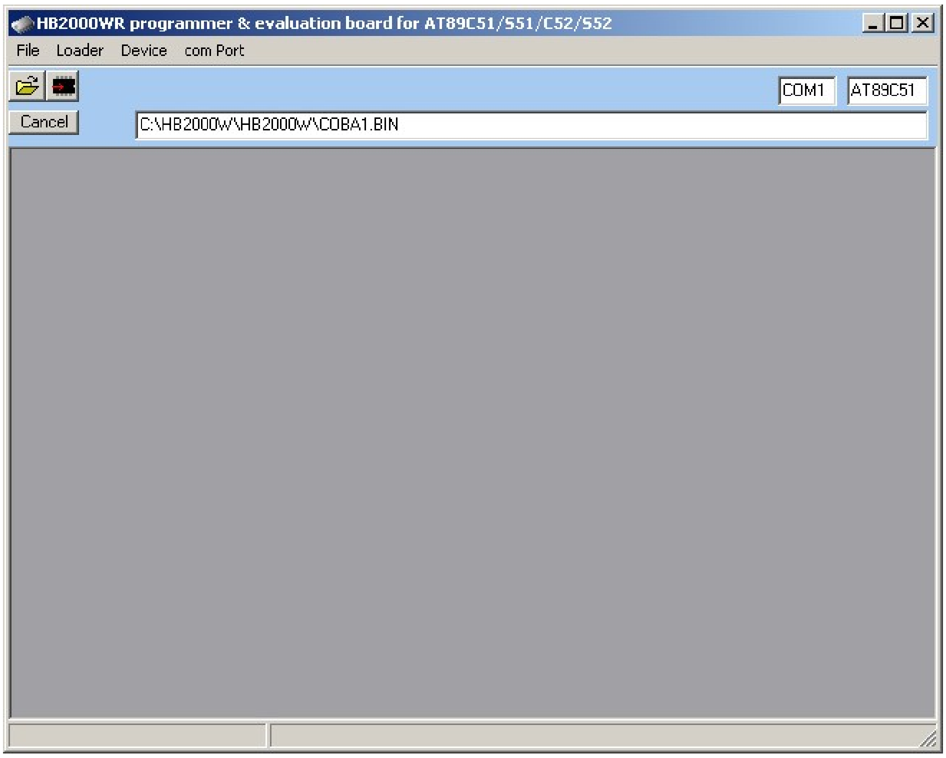
   Ada tiga macam instruksi yang dibutuhkan dalam pengalamatan kode yaitu : relative jump, in – blok jump atau call dan long jump atau call.

I.2. Pengenalan Program HB2000WR

HB2000WR merupakan perangkat lunak yang berisikan semua kebutuhan yang diperlukan dalam pengembangan mikrokontroler ATMEL AT89C51/AT89S51. Cara pengoperasian program HB2000WR sebagai berikut:

Untuk mengaktifkan program, klik dua kali pada icon HB2000WR. Program ini digunakan mengisi file yang dicompile oleh program editor RIDE -51. Pastikan koneksi modul trainer dengan PC dan power supply telah tersambung dengan benar.

Akan tetapi jika semuanya telah tersambung dengan benar maka akan muncul dilayar anda sbb:



Gambar 1-1

Tampilan HB2000WR

Penjelasan Menu :

1. Menu *File* berisi Open File dan Exit.

   Open File berfungsi untuk membuka file berekstensi *bin* yang akan diisikan ke mikrokontroller.

   Exit berfungsi untuk menutup program HB200WR.

2. Menu *Loader* berisi *Erase Device*, *Program Flash*, *Read Flash*, *LockBits*, *Auto Program*, *Verify*, *Blank Check*.

   Erase Device berfungsi untuk menghapus isi dari mikrokontroller.

   Program Flash untuk mengisi chip mikrokontroler dengan data dari file biner yang telah anda load atau dari hasil kompilasi program assembly.

Read Flash berfungsi untuk membaca isi memory mikrontroller.

LockBits berfungsi untuk melindungi program anda agar isi chip tidak bisa dibaca.

Auto Program digunakan untuk kompilasi secara otomatis dan dilanjutkan dengan proses pengisian program ke chip.

Verify digunakan untuk verifikasi program yang telah diisikan pada chip.

Blank Check berfungsi untuk mengecek chip apakah benar-benar telah kosong.

3. Menu *Device* berfungsi untuk memilih tipe chip mikrokontroler yang digunakan.
4. Menu *COM Port* berfungsi untuk memilih COM yang digunakan.

BAB II
PEMROGRAMAN PORT OUTPUT

II.1. Tujuan Percobaan

- Praktikan dapat mengirimkan data sederhana ke port

II.2. Dasar Teori

Mikrokontroler AT89C51/AT89S51 memiliki 4 buah port I/O yang terdiri dari :

- Port 0

  Port 0 merupakan port dua fungsi yang berada pada pin 32 - 39, pada rancangan system sederhana port ini digunakan sebagai port I/O serbaguna. Untuk rancangan yang lebih komplek yang melibatkan memori eksternal jalur ini dimultiplek untuk bus data dan bus alamat.

- Port 1

  Port 1 disediakan khusus sebagai port I/O, port ini berada pada pin 1-8.

- Port 2

  Port 2 (pin 21- 28) merupakan port dua fungsi yaitu sebagai port I/O serbaguna, dan sebagai bus alamat byte tinggi untuk rancangan system yang melibatkan memori eksternal.

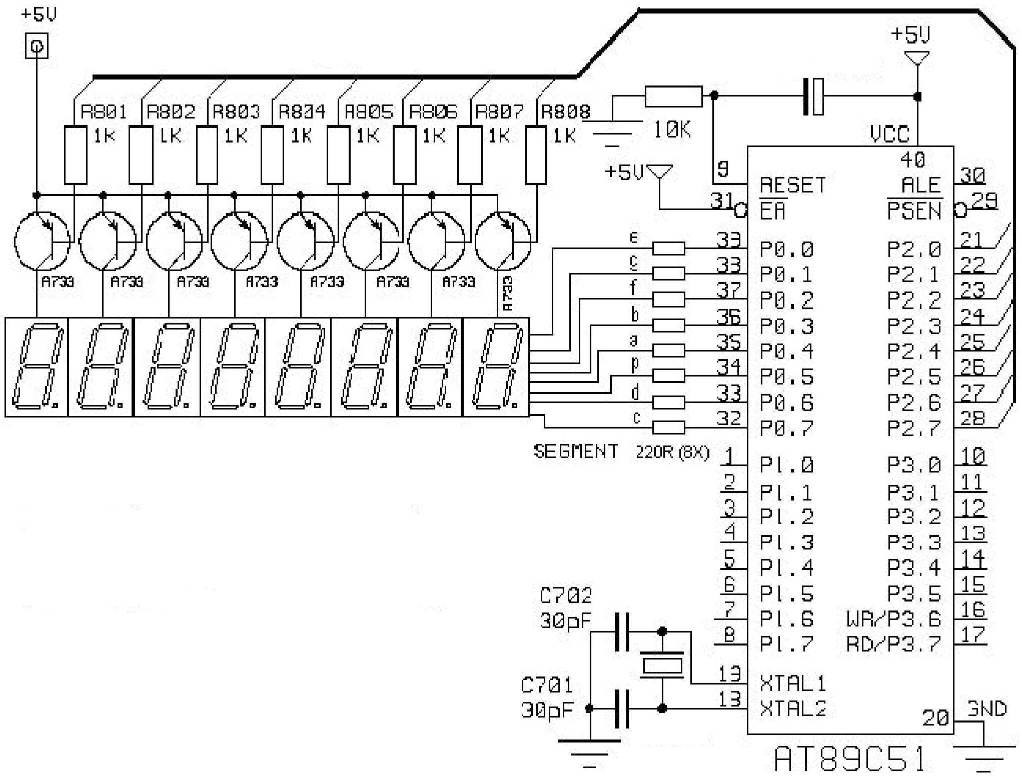
- Port 3

  Port 3 adalah port dua fungsi yang berada pada pin 10 – 17, port ini memiliki fungsi ganda seperti diuraikan pada table 1 berikut ini.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (extenal interrupt 0)
P3.3	$\overline{\text{INT1}}$ (extenal interrupt 1)
P3.4	T0 (timer 0 extenal input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (extenal data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Tabel 3-1

Port 3



Gambar 2-2

Rangkaian Display Seven Segment

II.3. Langkah Percobaan

II.3.1 Output ke Tampilan LED

1. Pastikan saklar Main Power dalam posisi ON dan lainnya dalam posisi OFF.
2. Pastikan koneksi ke PC sudah benar dan program RIDE-51 telah dijalankan.
3. Klik menu *Project* pilih *New*.Lalu pilih type chip dan buat nama File. Setelah itu klik *Terminate*.
4. Setelah itu masuk ke menu *File* lalu pilih *New >> Assembler File*.
5. Masuk ke menu Option >> Project >> LX51 >> Linker. Beri tanda pada check box *Generate a Binary File.*
6. Ketik program di bawah ini

```
$include(REG51.INC)

        org 0h
mulai:  mov     p1,#0fh
        call    tunda
        mov     p1,#0f0h
        call    tunda
```

```
          sjmp    mulai
tunda:    mov     r0,#0ffh
tunda1:   mov     r1,#0ffh
          djnz    r1,$
          djnz    r0,tunda1
          ret
          end
```

7. Pilih menu *Project* >> *Add Node Source/Application* lalu beri nama file dengan ekstensi .a51
8. Lakukan kompilasi tombol *Shift+F9*.
9. Lakukan pengisian dengan program HB2000WR.
10. Jalankan program dan lakukan pengamatan hasilnya pada display Binary dan Hexadesimal port yang bersangkutan. Set display led ke *active high*, dengan cara menekan tombol *Binary Led Option* sehingga mematikan led.
11. Ulangi percobaan dengan mengubah data sesuai instruksi.

II.3.2 Output ke Tampilan 7-Segment

12. Pastikan saklar Main Power dalam posisi ON dan lainnya dalam posisi OFF.
13. Pastikan koneksi ke PC sudah benar dan program RIDE-51 telah dijalankan.
14. Klik menu *Project* pilih *New*.Lalu pilih type chip dan buat nama File. Setelah itu klik *Terminate*.
15. Setelah itu masuk ke menu *File* lalu pilih *New* >> *Assembler File*.
16. Masuk ke menu Option >> Project >> LX51 >> Linker. Beri tanda pada check box *Generate a Binary File.*
17. Ketik program di bawah ini

```
;Program untuk menghidupkan display 8 X seven segment
;Untuk menampilkan tulisan Electro
;---------------------------------------------------------------------------
$include(REG-51)
        org 0h
mulai:  mov     dptr,#kata
        mov     r6,#08h
        mov     r1,#7fh
ulang:  clr     a
        movc    a,@a+dptr
        inc     dptr
        mov      p0,a
        mov     a,r1
```

```
        mov   p1,a
        rr    a
        mov   r1,a
        call  tunda
        djnz  r6,ulang
tunda:  mov   r2,#0ffh
        djnz  r2,$
        ret
        end
kata:
       db  0a8h,0bah,0a0h,0aah,0b8h,0eah,22h
```

18. Pilih menu *Project* >> *Add Node Source/Application* lalu beri nama file dengan ekstensi .a51
19. Lakukan kompilasi tombol *Shift+F9.*
20. Lakukan pengisian dengan program HB2000WR.
21. Jalankan program dan lakukan pengamatan hasilnya pada display 7 segment.
22. Ulangi percobaan dengan mengubah data sesuai instruksi.

II.4. Tugas dan Pertanyaan

1. Buatlah program untuk menampilkan nyala LED yang bergantian satu bit. Jelaskan cara kerja program tersebut.
2. Buatlah program untuk menuliskan NIM anda pada display seven segment. Jelaskan cara kerja program tersebut.

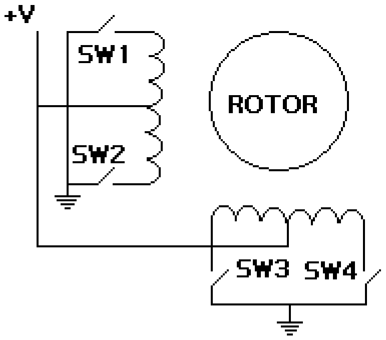
BAB III
MOTOR STEPPER

III.1. Tujuan Percobaan

- Praktikan dapat menggunakan mikrokontroller untuk menggerakkan motor stepper.
- Praktikan dapat mengatur arah dan kecepatan motor stepper.

III.2. Dasar Teori

Motor Stepper adalah motor yang bergeraknya per langkah, dan geraknya dikendalikan oleh data digital. Inti motor (rotor) berupa magnet permanen dan stator terdiri dari empat kumparan yang dihubungkan dengan saklar. Dengan membuka menutupnya saklar sesuai dengan data pada tabel 4, maka kita dapat memutar inti motor dengan arah tertentu. Adapun urutan data yang harus diberikan ditunjukkan pada tabel 4 dibawah ini. Untuk arah putaran kebalikkannya, maka urutan data pada tabel 4 harus dibalik.



Gambar 3-1

Motor Stepper

Step	SW1	SW2	SW3	SW4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Tabel 3-1

Data Motor Stepper

III.3. Langkah percobaan

III.3.1 Menggerakkan Motor Stepper

23. Pastikan saklar Main Power dalam posisi ON dan lainnya dalam posisi OFF.
24. Pastikan koneksi ke PC sudah benar dan program RIDE-51 telah dijalankan.
25. Klik menu *Project* pilih *New*.Lalu pilih type chip dan buat nama File. Setelah itu klik *Terminate*.
26. Setelah itu masuk ke menu *File* lalu pilih *New >> Assembler File*.
27. Masuk ke menu Option >> Project >> LX51 >> Linker. Beri tanda pada check box *Generate a Binary File.*
28. Ketik program di bawah ini

```
;PROGRAM UNTUK MENGGERAKKAN MOTOR STEPPER
;PUTARAN 360  KE ARAH KANAN
;---------------------------------------------------------------------------
$include(REG51.INC)
        org     0h
mulai:  mov     r2,#50
ulang:  mov     p1,#10h
        call    tunda
        mov     p1,#20h
        call    tunda
        mov     p1,#40h
        call    tunda
        mov     p1,#80h
        call    tunda
        djnz    r2,ulang
        jmp     $
tunda:  mov     r0,#0ffh
tunda1:mov r1,#50h
        djnz    r1,$
        djnz    r0,tunda1
        ret
        end
```

29. Pilih menu *Project* >> *Add Node Source/Application* lalu beri nama file dengan ekstensi .a51
30. Lakukan kompilasi tombol *Shift+F9.*
31. Lakukan pengisian dengan program HB2000WR.
32. Jalankan program dan lakukan pengamatan hasilnya pada display Binary dan Hexadesimal port yang bersangkutan. Set display led ke

active high, dengan cara menekan tombol Binary Led Option sehingga mematikan led.

33. Ulangi percobaan dengan mengubah data sesuai instruksi.

III.3.2 Pengaturan Arah Motor Stepper

34. Pastikan saklar Main Power dalam posisi ON dan lainnya dalam posisi OFF.
35. Pastikan koneksi ke PC sudah benar dan program RIDE-51 telah dijalankan.
36. Klik menu *Project* pilih *New*.Lalu pilih type chip dan buat nama File. Setelah itu klik *Terminate*.
37. Setelah itu masuk ke menu *File* lalu pilih *New >> Assembler File*.
38. Masuk ke menu Option >> Project >> LX51 >> Linker. Beri tanda pada check box *Generate a Binary File.*
39. Ketik program di bawah ini.

```
;PROGRAM UNTUK MENGGERAKKAN MOTOR STEPPER
;PUTARAN 360  KE ARAH KANAN
;-------------------------------------------------------------------------
$include(REG51.INC)
            org  0h
mulai:      mov  r2,#50
ulang:      mov  p1,#80h
            call tunda
            mov  p1,#40h
            call tunda
            mov  p1,#20h
            call tunda
            mov  p1,#10h
            call tunda
            djnz r2,ulang
            jmp        $
tunda:      mov  r0,#0ffh
tunda1:     mov  r1,#50h
            djnz r1,$
            djnz r0,tunda1
            ret
            end
```

40. Pilih menu *Project >> Add Node Source/Application* lalu beri nama file dengan ekstensi .a51
41. Lakukan kompilasi tombol *Shift+F9.*
42. Lakukan pengisian dengan program HB2000WR.

43. Posisikan saklar Stepper Motor Driver Module pada posisi ON
44. Jalankan program dan lakukan pengamatan hasilnya pada display pergerakan motor stepper.
45. Ulangi percobaan dengan mengubah data sesuai instruksi.

III.3.3 Pengaturan Kecepatan Motor Stepper

46. Pastikan saklar Main Power dalam posisi ON dan lainnya dalam posisi OFF.
47. Pastikan koneksi ke PC sudah benar dan program RIDE-51 telah dijalankan.
48. Klik menu *Project* pilih *New*.Lalu pilih type chip dan buat nama File. Setelah itu klik *Terminate*.
49. Setelah itu masuk ke menu *File* lalu pilih *New* >> *Assembler File*.
50. Masuk ke menu Option >> Project >> LX51 >> Linker. Beri tanda pada check box *Generate a Binary File.*
51. Ketik program di bawah ini

```
;PROGRAM UNTUK MENGGERAKKAN MOTOR STEPPER
;PUTARAN 360  KE ARAH KANAN
;--------------------------------------------------------------------------------------
$include(REG51.INC)
            org  0h
mulai:      mov        r2,#50
ulang:      mov        p1,#10h
            call  tunda
            call  tunda
            mov        p1,#20h
            call  tunda
            call  tunda
            mov        p1,#40h
            call  tunda
            call  tunda
            mov        p1,#80h
            call  tunda
            call  tunda
            djnz  r2,ulang
            jmp        $
tunda:      mov  r0,#0ffh
tunda1:     mov  r1,#50h
            djnz  r1,$
            djnz  r0,tunda1
            ret
            end
```

52. Pilih menu *Project* >> *Add Node Source/Application* lalu beri nama file dengan ekstensi .a51
53. Lakukan kompilasi tombol *Shift+F9*.
54. Lakukan pengisian dengan program HB2000WR.
55. Posisikan saklar Stepper Motor Driver Module pada posisi ON.
56. Jalankan program dan lakukan pengamatan hasilnya pada pergerakan motor stepper.
57. Ulangi percobaan dengan mengubah data sesuai instruksi.

III.4. Tugas dan Pertanyaan

1. Buatlah program untuk menggerakkan motor stepper 45° searah jarum jam. Jelaskan cara kerja program tersebut.
2. Buatlah program untuk menggerakkan motor stepper 180° searah jarum jam dan kembali ke posisi awal dan diulang terus-menerus. Jelaskan cara kerjanya.

BAB IV
PEMROGRAMAN PORT INPUT

IV.1. Tujuan Percobaan

- Praktikan dapat memahami cara membaca data dari input keypad 4x4

IV.2. Dasar Teori

Untuk membaca data dari keypad 4X4 adalah sebagai berikut :

- Untuk mendeteksi tombol pada suatu kolom maka kolom tersebut harus dibuat logika ‘0’ dan kolom yang lainya harus dibuat logika ‘1’.
- Untuk mendeteksi penekanan tombol pada suatu baris maka deteksi melalui bit port yang terhubung pada baris tersebut.

IV.3. Langkah Percobaan

IV.3.1 Menggerakkan Motor Stepper dengan Keypad 4x4

58. Pastikan saklar Main Power dalam posisi ON dan lainnya dalam posisi OFF.
59. Pastikan koneksi ke PC sudah benar dan program RIDE-51 telah dijalankan.
60. Klik menu *Project* pilih *New*.Lalu pilih type chip dan buat nama File. Setelah itu klik *Terminate*.
61. Setelah itu masuk ke menu *File* lalu pilih *New* >> *Assembler File*.
62. Masuk ke menu Option >> Project >> LX51 >> Linker. Beri tanda pada check box *Generate a Binary File.*
63. Ketik program di bawah ini

```
;-----------------------------------------------------------------
;Program membaca data dari keypad 4X4 baris I
;Untuk menjalankan motor stepper
;Pilih 1 untuk memutar motor stepper 360 ke kiri
;Pilih 2 untuk memutar motor stepper 360 ke kanan
;Pilih 3 untuk memutar motor stepper 180 ke kiri
;Pilih A untuk memutar motor stepper 180 ke kanan
;------------------------------------------------------------------
$include(REG51.INC)
            org 0h
```

```
mulai:    mov     p2,#11101111b
          jb      p2.0,pilih
          mov     r2,#50
          call    kanan
pilih:    jb      p2.1,pilih1
          mov     r2,#50
          call    kiri
pilih1:   jb      p2.2,pilih2
          mov     r2,#25
          call    kanan
pilih2:   jb      p2.3,mulai
          mov     r2,#25
          call    kiri
          jmp     mulai
kanan:    mov     p1,#10h
          call    tunda
          mov     p1,#20H
          call    tunda
          mov     p1,#40h
          call    tunda
          mov     p1,#80h
          call    tunda
          djnz    r2,kanan
          ret
kiri:     mov     p1,#80h
          call    tunda
          mov     p1,#40H
          call    tunda
          mov     p1,#20h
          call    tunda
          mov     p1,#10h
          call    tunda
          djnz    r2,kiri
          ret
tunda:    mov     r0,#0ffh
tunda1:   mov     r1,#50h
          djnz    r1,$
          djnz    r0,tunda1
          ret
          end
```

64. Pilih menu *Project >> Add Node Source/Application* lalu beri nama file dengan ekstensi .a51
65. Lakukan kompilasi tombol *shift+F9.*
66. Lakukan pengisian dengan program HB2000WR.
67. Posisikan saklar Stepper Motor Driver Module pada posisi ON.
68. Jalankan program dan lakukan pengamatan pada pergerakan motor stepper.

69. Ulangi percobaan dengan mengubah data sesuai instruksi.

IV.4. Tugas dan Pertanyaan

1. Buatlah program dengan ketentuan sbb:
   - Jika pada keypad ditekan angka 1 maka motor stepper akan bergerak searah jarum jam 2x360^o dan berhenti.
   - Jika pada keypad ditekan angka 2 maka motor stepper akan bergerak searah jarum jam 90° dan berhenti.
   - Jika pada keypad ditekan angka 3 maka motor stepper akan bergerak berlawanan arah jarum jam 180° dan berhenti.

   Jelaskan cara kerja program tersebut.

BAB V
TAMPILAN LCD

V.1. Tujuan Percobaan

V.2. Dasar Teori

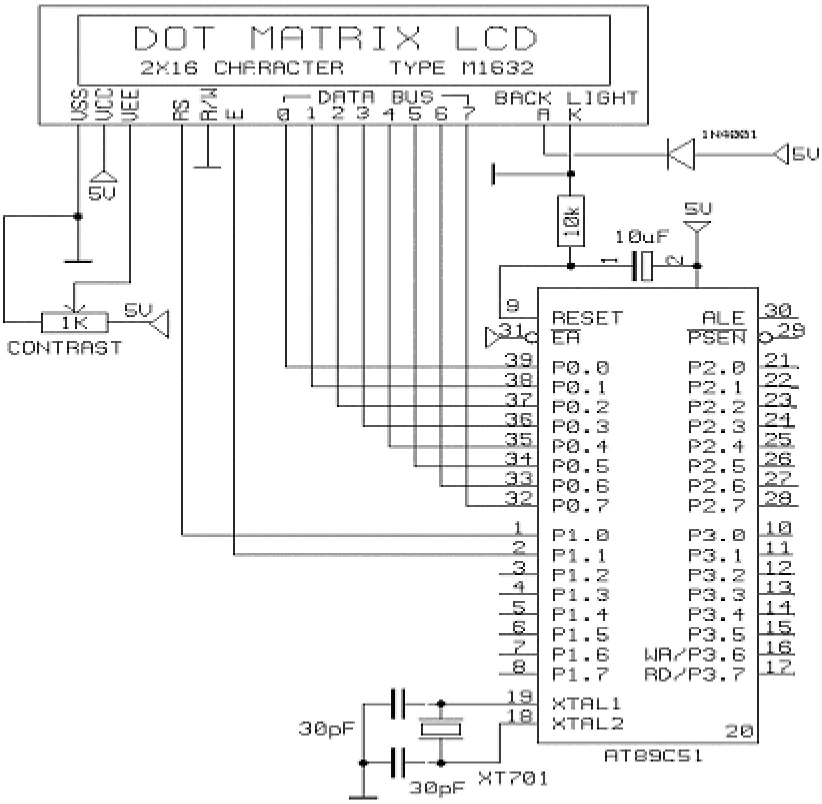
LCD yang dipergunakan pada module ini adalah LCD tipe M1632 yaitu sebuah LCD dot matrix16X2 baris dengan konsumsi daya rendah. Adapun fasilitas yang dimiliki meliputi :

- 16 karakter, 2 baris LCD dot matrik 5X7 + *cursor*
- Karakter generator ROM untuk 192 karakter
- Data RAM Display maksimum 80 karakter
- Memungkinkan antarmuka dengan 4 bit data atau 8 bit data
- Memiliki beberapa instruksi :
  - o Display clear
  - o Cursor home
  - o Display ON/OFF
  - o Cursor ON/OFF
  - o Display Character Blink
  - o Cursor Shift
  - o Display Shift
- + 5V single power supply

Nama Signal	No. Terminal	I/O	Tujuan	Fungsi
DB0-DB3	4	I/O	MPU	Data bus 4 bit bawah *(tristate bi-directional)*: Bus data ini dipakai untuk membaca atau menulis data. Jika interface data menggunakan 4 bit maka signal ini tidak digunakan.
DB4-DB7	4	I/O	MPU	Data bus 4 bit atas (*tristate bi-directional*): Bus data ini dipakai untuk membaca atau menulis data. DB7 juga digunakan untuk *busy flag*.
E	1	I	MPU	Enable Signal Signal untuk mengaktifkan tulis data atau baca data

R/W	1	I	MPU	Signal pemilih mode read atau write 0 : write 1 : read
RS	1	I	MPU	Register selection signal 0 : Instruction register (read) 1 : Data register (write and read)

Tabel 5-1
Tabel Instruksi LCD



Gambar 5-1
Rangkaian LCD

Kode	*HEX*	*Kode*	*HEX*	*Kode*	*HEX*	*Kode*	*HEX*
0	22H	A	60H	H	70H	t	0B8H
1	77H	a	24H	h	78H	U	32H
2	0A4H	b	38H	I	77H	y	31H
3	25H	C	0AAH	J	37H	-	0FDH
4	71H	d	34H	L	0BAH	.	0DFH
5	29H	E	0A8H	n	62H	BLANK	0BDH
6	28H	e	0A0H	O	22H		
7	67H	F	E8H	P	0E0H		
8	20H	G	28H	r	0EAH		
9	21H	g	21H	S	29H		

Tabel 5-2

Kode Alpha Numerik Display Seven Segment

V.3. Langkah Percobaan

V.3.1 Menampilkan Data di LCD

70. Pastikan saklar Main Power dalam posisi ON dan lainnya dalam posisi OFF.
71. Pastikan koneksi ke PC sudah benar dan program RIDE-51 telah dijalankan.
72. Klik menu *Project* pilih *New*.Lalu pilih type chip dan buat nama File. Setelah itu klik *Terminate*.
73. Setelah itu masuk ke menu *File* lalu pilih *New >> Assembler File*.
74. Masuk ke menu Option >> Project >> LX51 >> Linker. Beri tanda pada check box *Generate a Binary File.*
75. Ketik program di bawah ini

```
;=========================================
; Program demo menampilkan tulisan pada LCD
;=========================================
$include(REG51.INC)
LCD_RS          bit     P1.0
LCD_CS          bit     P1.1

                org     0h
mulai:          call    inisialisasi
                mov     dptr,#tulis_atas
                mov     r3,#16
                mov     a,#80h
                call    write_inst
tulis:          clr     a
                movc    a,@a+dptr
                inc     dptr
                call    write_data
                djnz    r3,tulis
```

```
                mov     dptr,#tulis_bawah
                mov     r3,#16
                mov     a,#0C0h
                call    write_inst
tulis1:         clr     a
                movc    a,@a+dptr
                inc     dptr
                call    write_data
                djnz    r3,tulis1
                jmp     mulai
;=================
; Inisialisasi LCD
;=================
inisialisasi:
                mov     A,#03Fh
                call    write_inst
                call    write_inst
                mov     A,#0Dh
                call    write_inst
                mov     A,#06h
                call    write_inst
                mov     A,#01h
                call    write_inst
                mov     A,#0Fh
                call    write_inst
                ret
;=====================================
; Routine untuk menulis instruksi ke LCD
;=====================================
write_inst:
                clr     LCD_RS
                setb    LCD_CS
                mov     P0,A    ;intruksi ke LCD
                clr     LCD_CS  ;module
                setb    LCD_CS
                call    delay
                ret
;=====================================
; Routine untuk menulis data ke LCD
;=====================================
write_data:
                setb    LCD_RS
                setb    LCD_CS
                mov     P0,A    ;data ke LCD
                clr     LCD_CS  ;module
                setb    LCD_CS
                call    delay
                ret
;=======================
; Routine penghasil delay
;=======================
delay:          mov     R0,#0FFh
```

```
delay1:     mov    R5,#0
            djnz   R5,$
            djnz   R0,delay1
            ret
;
Ldelay:     mov    R2,#010h
Ld1:        call   delay
            djnz   R2,Ld1
            ret
;
tulis_atas:
             db 'Teknik Elka D3  '
tulis_bawah:
             db 'ITN Malang      '
             end
```

76. Pilih menu *Project >> Add Node Source/Application* lalu beri nama file dengan ekstensi .a51
77. Lakukan kompilasi tombol *Shift+F9.*
78. Lakukan pengisian dengan program HB2000WR.
79. Posisikan saklar Stepper Motor Driver Module pada posisi ON.
80. Jalankan program dan lakukan pengamatan pada layar LCD.
81. Ulangi percobaan dengan mengubah data sesuai instruksi.

V.4. Tugas dan Pertanyaan

1. Buat program untuk menuliskan nama dan NIM anda pada baris 1 dan dilanjutkan geser tulisan ke kiri! Jelaskan program tersebut!

TATA TERTIB

PELAKSANAAN PRAKTIKUM MIKROKONTROLLER

1. Setiap Praktikan di wajibkan memiliki **Buku Petunjuk Praktikum**
2. 15 (Lima belas) menit sebelum praktikum dimulai Praktikan harus sudah hadir di **Laboratorium Pemeliharaan Dan Perbaikan Perancangan Elektonika** dan menyerahkan **Lembar Asistensi** kepada Instruktur / asisten yang bersangkutan.
3. Praktikan diharuskan berpakaian rapi serta sopan (tidak diperkenankan mengenakan **Kaos Oblong** dan bersepatu sandal pada waktu praktikum.
4. Praktikan menyediakan sendiri alat-alat tulis / gambar yang di perlukan.
5. Selama praktikum berlangsung, praktikan di larang merokok dan diwajibkan ikut menjaga ketertiban suasansa jalanya praktikum.
6. Praktikum di mulai pagi hari pukul 07.30 s.d. selesai.
7. Terlambat hadir selama 30 menit dianggap mengundurkan diri dan biaya praktikum tidak di kembalikan.
8. Tidak hadir selama praktikum dianggap mengundurkan diri dan biaya praktikum tidak di kembalikan.
9. Untuk tiap percobaan/judul praktikum disediakan tempat atau peralatan /instrument sendiri, yang tidak boleh ditukar atau diganti kecuali Assisten yang bersangkutan.
10. Sebelum mengerjakan praktikum, praktikan diwajibkan menguasai hal-hal yang akan di praktikumkan (teori yang berhubungann dengan yang akan di lakukan/dilaksanakan) dan mengerjakan tugas yang diberikan pada setiap judul praktikum (baca petunjuk dan jadwal praktikum untuk masing-masing kelompok).
11. Setiap kali sebelum praktikum di mulai, Praktikan akan diberi penjelasan oleh Asisten tentang butir 9.
12. Bila menjumpai kesalahan atau terjadi ketidak cocokan dengan buku petunjuk agar segera melapor pada Asisten.
13. Praktikan yang merusakkan barang (mesin maupun alat - alat) di bawah Rp. 10.000,- (sepuluh ribu rupiah) harus mengganti sesuai dengan harga atau berupa barang yang dirusakkan, sedangkan bila kerusakan di atas nilai / harga tersebut, maka ditanggung oleh satu regu / kelompok praktikum dengan ketentuan sebagai berikut :
    - Untuk praktikan yang merusakkan, mengganti 50% dari harga.
    - Sedang sisanya, ditanggung oleh satu regu/kelompok.
14. Setiap selesai melaksanakan praktikum, praktikan diwajibkan mengembalikan peralatan yang telah digunakan dan dilarang meninggalkan ruang praktikum sebelum mendapat ijin dari asisten yang bersangkutan.
15. Setelah praktikum selesai, setiap praktikum harus mendapat persetujuan dari asisten.
16. Pratikan harus mengerjakan tugas sesuai format yang sudah di berikan dan ditetapkan oleh Lab.

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.